

DOKUMENTACE K PROJEKTU DATABÁZOVÝ SYSTÉM

Jméno autora: Jiří Laštovka

Třída: C4c

Datum vypracování: 02.02.2023 – 03.02.2023

Účel projektu: školní projekt

Škola: Střední průmyslová škola elektrotechnická, Praha 2, Ječná 30

Adresa školy: Ječná 30, 120 00, Praha 2

Požadavky na práci s programem:

Pro zajištění ideálních podmínek ke spuštění programu by uživatel měl vlastnit program PyCharm, ve kterém lze program spustit. Instalace zde: <https://www.jetbrains.com/pycharm/download>

Popis programu:

Program slouží k ovládání databáze – ke vkládání, úpravě, mazání a zobrazování dat. Program je snadno rozšiřitelný díky separaci jednotlivých částí kódu do modulů.

Ovládání programu:

Program se ovládá pouze pomocí klávesnice. Uživatel je veden programem pomocí výpisů na obrazovku. Vždy po zadání vstupu se volba potvrzuje klávesou ENTER.

Architektura programu:

Snažil jsem se o co možná největší přiblížení se k architektuře Three-Tier. Metoda main() je ‘presentation tier’, který komunikuje s uživatelem, moduly pojmenovány po databázových objektech (film, promítání, ...) jsou ‘application tier’, jelikož se v nich data zpracovávají a formátují. Moduly končící na DAO (Data Access Object) jsou ‘data tier’. Získávají totiž data z databáze a také data vkládají – provádějí SQL příkazy. Hlavní argumentem pro použití této architektury je separace jednotlivých funkcí a modulů podle jejich funkcionality. Například pokud je v metodě main() potřeba vypsat informace o kinosálech v databázi, zavolá se funkce get_all_kinosal() z modulu kinosal.py, která zavolá funkci select_all_kinosal() z modulu kinosalDAO.py. Nikdy se v main() nevolají funkce z modulů končících na DAO.

Tento program nepoužívá pojem DAO v původním slova smyslu. Nevytváří se tedy žádný objekt, který by představoval daný objekt v databázi. V tomto programu se používá koncovka DAO pouze pro odlišení vrstev ‘application tier’ a ‘data tier’.

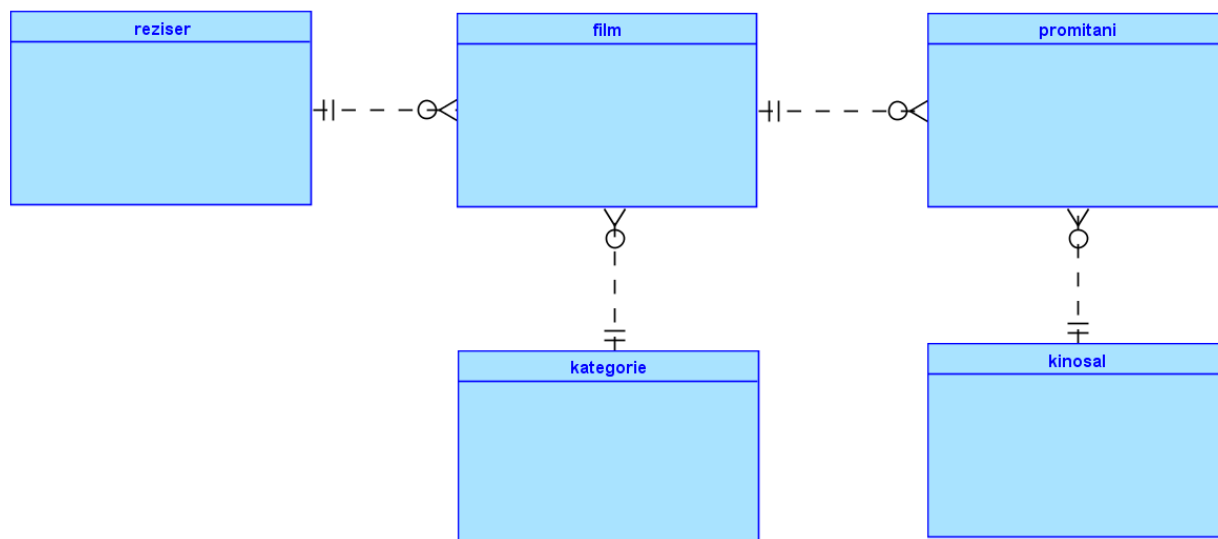
Popis struktury programu:

Program obsahuje soubor main.py, který skrze metodu main() řídí program. V této metodě main() se používají metody z modulů, které jsou pojmenovány po databázových objektech (například kinosál či film). Jsou to metody, které získávají data z dalších modulů DAO, a následně je pomocí funkce format_data() z modulu lib/data_formating.py připraví na zobrazení uživateli. Metody v modulech končících na DAO jsou pojmenovány po SQL příkazech odpovídajících jejich funkcionality (například insert_reziser()). Tyto metody se pomocí funkce get_instance() z modulu database_singleton_connection.py napojí do databáze a provádějí SQL příkazy. Pokud získají nějaká data, vracejí je pomocí return. Samotná statická metoda get_instance() se nachází ve třídě DatabaseConnectionSingleton. Ve složce lib se nachází kód z internetu s pomůckou pro formátování dat z databáze do přehledných textových tabulek. Dále program obsahuje konfigurační soubor ve formátu json config.json, ve kterém lze nastavit databázi, do které se program připojuje, server, uživatelské jméno a heslo. Předpokladem ale je, že v této databázi jsou již vytvořeny tabulky se kterými program pracuje. Konfigurace ze souboru config.json se v programu načítá funkcí load_config() v souboru configuration.py. V této funkci se také kontroluje správnost této konfigurace.

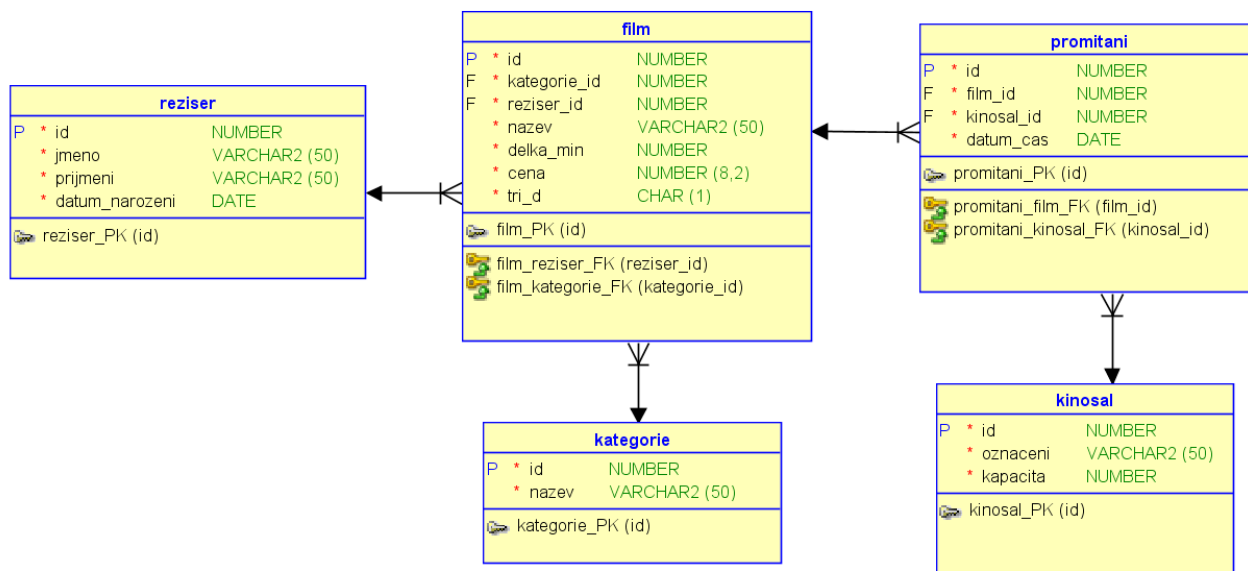
Chod aplikace:

Po spuštění se objeví hlavní menu, které obsahuje možné akce. Uživatel vybere akci pomocí klávesnice a potvrdí enterem. Program podle volby provede uvedenou akci. Například při výběru '6' se objeví tabulka s výpisem všech filmů v databázi. Po provedené akci se program vrátí na hlavní menu. Při jakkoli chybně zadaném vstupu program uživatele upozorní na chybu a vyžaduje opravu, dokud vstup není validní. Pro ukončení programu uživatel v hlavním menu zvolí možnost '20', čímž se program ukončí.

Logické schéma databáze:



Relační schéma databáze:



Kód třetí strany:

Program obsahuje složku lib, ve které je modul data_formating.py. Tento soubor importuje knihovnu tabulate, která formátuje data do textu. V tomto souboru jsem použil funkci format_data(), která vytvoří z 2D pole tabulku.

Chybové stavy programu:

Tento program všechny chyby řeší chybovou hláškou bez zastavení programu a vyzývá uživatele k opravě vstupu. Chybová hláška uživateli napoví, kde se stala chyba.

Rozšiřitelnost programu:

Tento program lze snadno rozšiřovat například přidáváním nových funkcionalit a modulů. Moduly DAO jsou připraveny k používání moduly z druhé vrstvy. Ne všechny funkce z modulů DAO jsou využity. V tom spočívá rozšiřitelnost programu.